This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method for logging while updating a B-link tree via a plurality of data transactions, comprising:

generating a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on a plurality of nodes of a B-link tree, wherein the data transactions are associated with a single B-link tree operation on said B-link tree, wherein said plurality of log entries include at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

storing said plurality of log entries in a single log, the single log being a partially persistent log, wherein a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage;

associating said plurality of log entries with each other for ~~use in~~ determining whether the single B-link tree operation has been completed; and

wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time.

2. (Original) A method according to claim 1, further including periodically truncating the log.

3. (Canceled)

4. (Previously Presented) A method according to claim 1, further including discarding one of the log entries from the log when the corresponding data transaction has been carried out on said B-link tree.

5. (Previously Presented) A method according to claim 1, wherein said storing includes storing said log entries into the log before the corresponding data transactions are carried out on said B-link tree.

6. (Previously Presented) A method according to claim 1, further including caching data of said data transactions before said data transactions are carried out on said B-link tree.

7 - 8. (Canceled)

9. (Previously Presented) A method according to claim 1, further including maintaining a log sequence number with each of said plurality of log entries, uniquely identifying one of the log entries.

10. (Previously Presented) A method according to claim 1, wherein said method is embodied in instructions that are stored on a computer readable medium in physical memory.

11. (Canceled)

12. (Currently Amended) A method for logging while updating a B-link tree via a plurality of data transactions, comprising:

generating a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on a plurality of nodes of a B-link tree, wherein the data transactions are associated with a single B-link tree operation on said B-link tree, wherein said plurality of log entries include at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

storing said plurality of log entries in a single log, the single log being a partially persistent log, wherein a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage;

associating said plurality of log entries with each other for ~~use in~~ determining whether the single B-link operation has been completed; and

wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time.

periodically flushing data corresponding to data transactions represented by the finite log to persistent storage; and

truncating said finite log in coordination with said flushing.

13. (Canceled)

14. (Previously Presented) A method according to claim 12, further including discarding one of the log entries from the finite log when the corresponding data transaction has been carried out on said B-link tree.

15. (Previously Presented) A method according to claim 12, wherein said storing includes storing said log entries into the finite log before the corresponding data transactions are carried out on said B-link tree.

16. (Previously Presented) A method according to claim 12, further including caching data of said data transactions before said data transactions are carried out on said B-link tree.

17-18. (Canceled)

19. (Previously Presented) A method according to claim 12, further including maintaining a log sequence number with each of said plurality of log entries, uniquely identifying one of the log entries.

20. (Previously Presented) A method according to claim 12, wherein said method is embodied in instructions that are stored on a computer readable medium in physical memory.

21. (Canceled)

22. (Currently Amended) A method for logging while updating a data structure via a plurality of data transactions, comprising:

replicating updates to the data structure to a first server location and a second server location;

generating a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on said data structure, wherein the data transactions are associated with a single operation on the data structure; and

maintaining a single log, where the single log is partitioned into an upper layer and an allocation layer, at each of said first and second server locations, wherein the single log includes log entries from both the upper layer and allocation layer, wherein said entries from said allocation layer are local to their corresponding computing devices of a plurality of computing devices and said entries from said upper layer are replicated among said plurality of computing devices;

associating said plurality of log entries with each other for ~~use in~~ determining whether the single operation on the data structure has been completed; and

wherein said single log is a partially persistent log, where a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage.

23. (Original) A method according to claim 22, further including recovering the data structure after a failure by performing parallel recovery operations by each of said first and second server locations.

24. (Original) A method according to claim 22, wherein said data structure is a B-link tree.

25. (Original) A method according to claim 24, wherein the upper layer is a B-link tree layer that handles B-link tree operations.

26. (Previously Presented) A method according to claim 22, wherein the allocation layer handles at least one of (A) an allocate disk space operation, (B) a deallocate disk space operation,

(C) a read from the allocated disk space operation and (D) a write to the allocated disk space operation.

27. (Previously Presented) A method according to claim 22, wherein said method is embodied in instructions that are stored on a computer readable medium in physical memory.

28. (Canceled)

29. (Currently Amended) A server for maintaining a log while updating a B-link tree via a plurality of data transactions, comprising:

a logging object that generates a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on a plurality of nodes of a B-link tree, wherein the data transactions are associated with a single B-link tree operation on said B-link tree;

an allocation layer object for said B-link tree;

a B-link tree layer object, wherein said plurality of log entries include at least one entry from the allocation layer object and at least one entry from the B-link tree layer object, wherein said at least one entry from said allocation layer object is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer object is replicated among said plurality of computing devices; and

a single storage log for storing said plurality of log entries, wherein said storage log is a partially persistent log, wherein a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage, where said plurality of log entries are associated with each other for ~~use in~~ determining whether the single B-link tree operation has been completed; and

wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time.

30. (Previously Presented) A server according to claim 29, wherein the storage log including said plurality of log entries is periodically truncated.

31. (Canceled)

32. (Previously Presented) A server according to claim 29, wherein said plurality of log entries are discarded from the storage log when the data transactions have been carried out on said B-link tree.

33. (Previously Presented) A server according to claim 29, wherein said the log entries are each stored in the storage log before corresponding data transactions are carried out on said B-link tree.

34. (Previously Presented) A server according to claim 29, wherein data of said data transactions are cached in a cache memory before said data transactions are carried out on said B-link tree.

35-36. (Canceled)

37. (Previously Presented) A server according to claim 29, wherein said logging object generates a log sequence number with each of said plurality of log entries, uniquely identifying each of said plurality of log entries.

38. (Currently Amended) A server for logging while updating a B-link tree via a plurality of data transactions, comprising:

a first object for generating a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on a plurality of nodes of a B-link tree, wherein the data transactions are associated with a single B-link tree operation on said B-link tree;

an allocation layer and a B-link tree layer, wherein said plurality of log entries include at least one entry from the allocation layer and at least one entry from the B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

a single finite storage log for storing said plurality of log entries wherein said finite storage log is a partially persistent log, comprising of a memory and a persistent storage, wherein the plurality of log entries are associated with each other for ~~use in~~ determining whether the single B-link tree operation has been completed;

wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time;

a second object for periodically flushing data corresponding to data transactions represented by the plurality of log entries in the finite storage log to persistent storage, wherein said persistent storage is configured to receive and store said data after said data transactions commit; and

a third object for truncating said finite storage log in coordination with the operation of the flushing of the second object.

39. (Canceled)

40. (Previously Presented) A server according to claim 38, wherein one of the log entries is discarded from the finite storage log when the corresponding data transaction has been carried out on said B-link tree.

41. (Previously Presented) A server according to claim 38, wherein said log entries are stored in the finite storage log before the corresponding data transactions are carried out on said B-link tree.

42. (Previously Presented) A server according to claim 38, wherein data of each of said data transactions is cached in a cache memory before said the each of said data transactions is carried out on said B-link tree.

43. (Previously Presented) A server according to claim 38, further including storing said plurality of log entries in an intermediate memory previous to storing said plurality of log entries in the finite storage log.

44. (Previously Presented) A server according to claim 43, wherein said plurality of log entries are moved from intermediate memory to the finite storage log after the data transactions commit.

45. (Previously Presented) A server according to claim 38, wherein said first object generates a log sequence number with each of said plurality of log entries, uniquely identifying said each of the log entries.

46. (Currently Amended) A server for logging while updating a data structure via a plurality of data transactions, comprising:

a replication object that replicates updates to the data structure to a first server location and a second server location;

a logging object that generates a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on said data structure, wherein the data transactions are associated with a single operation on said data structure; and

a storage element within which a single log is maintained, wherein the single log is partitioned into an upper layer and an allocation layer, at each of said first and second server locations, and wherein the single log includes log entries from both the upper layer and allocation layer, wherein said entries from the upper layer are replicated to both said first server and said second server, while said entries from said allocation layer are stored locally on said first server and said second server, wherein said plurality of log entries are associated with each other for ~~use in~~ determining whether the single operation on the data structure has been completed;

wherein said single log is a partially persistent log, wherein a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage; and

wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time.

47. (Original) A server according to claim 46, wherein the data structure is recovered after a failure via parallel recovery operations by each of said first and second server locations.

48. (Original) A server according to claim 46, wherein said data structure is a B-link tree.

49. (Original) A server according to claim 48, wherein the upper layer is a B-link tree layer that handles B-link tree operations.

50. (Previously Presented) A server according to claim 46, wherein the allocation layer handles at least one of (A) an allocate disk space operation, (B) a deallocate disk space operation, (C) a read from the allocated disk space operation and (D) a write to the allocated disk space operation.

51. (Currently Amended) A computing device for logging while updating a B-link tree via a plurality of data transactions, comprising:

means for generating a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on a plurality of nodes of a B-link tree, wherein the data transactions are associated with a single B-link tree operation on said B-link tree, wherein said plurality of log entries include at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices; and

means for storing said plurality of log entries into a <u>single</u> log after said data transaction commits, wherein said log is a partially persistent log, wherein a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage;

means for associating said plurality of log entries with each other for ~~use in~~ determining whether the single B-link tree operation has been completed;

wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time.

52. (Original) A computing device according to claim 51, further including means for truncating the log periodically.

53. (Currently Amended) A computing device for logging while updating a B-link tree via a plurality of data transactions, comprising:

means for generating a plurality of log entries corresponding to a plurality of data transactions, the data transactions to be carried out on a plurality of nodes of a B-tree, wherein the data transactions are associated with a single B-link operation on said B-link tree, wherein said plurality of log entries include at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

means for storing said plurality of log entries into a single finite log, wherein said finite log is a partially persistent log, wherein a tail of said partially persistent log is in a memory, and said partially persistent log comprises of said memory and a persistent storage, and wherein a boundary between said memory and said persistent storage in said partially persistent log varies over time;

means for associating said plurality of log entries with each other for ~~use in~~ determining whether the single B-link operation has been completed;

means for periodically flushing data corresponding to data transactions represented by the finite log to persistent storage; and

means for truncating said finite log in coordination with said means for periodically flushing.

54. (Original) A computing device according to claim 53, further including means for discarding one of the log entries from the finite log when the corresponding data transaction has been carried out on said B-link tree.

55. (Currently Amended) A computing device for logging while updating a data structure via a plurality of data transactions, comprising:

means for replicating updates to the data structure to a first server location and a second server location;

means for generating a plurality of log entries corresponding to a plurality of data transactions, the data transaction to be carried out on said data structure, wherein the data transactions are associated with a single operation on the data structure; and

means for maintaining a single log, where the log is partitioned into an upper layer and an allocation layer, at each of said first and second server locations, wherein the single log includes log entries from both the upper layer and allocation layer, wherein said entries from the upper layer are replicated to both said first server and said second server, while said entries from said allocation layer are stored locally on said first server and said second server;

means for associating said plurality of log entries with each other for ~~use in~~ determining whether the single operation on the data structure has been completed; and

wherein said single log is a partially persistent log that has a boundary that changes over time between a persistent and non-persistent memory.

56. (Original) A computing device according to claim 55, wherein said data structure is recoverable after a failure by performing parallel recovery operations by each of said first and second server locations.